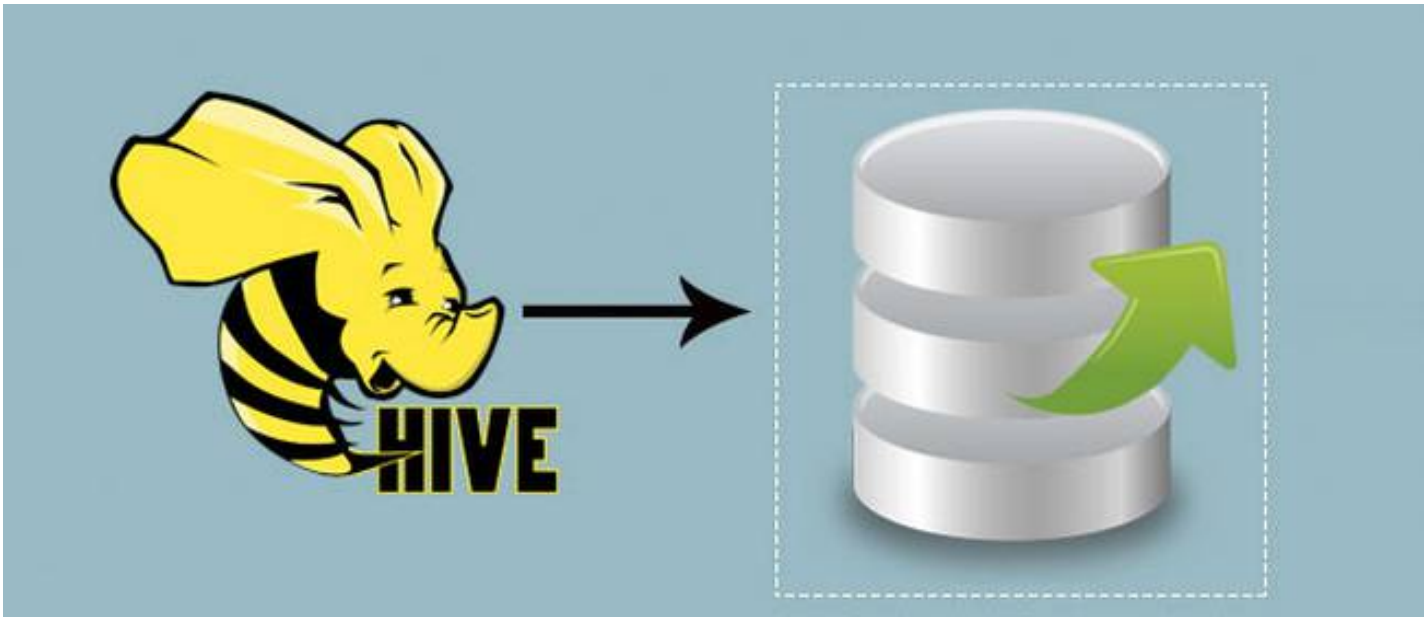
Hive SQL 查询函数手册



更新时间：2019-11-18 15:26:51　标签：`hive` `sql` 大数据 数据分析

Hive 提供了较完整的 SQL 功能，HQL 与 SQL 基本上一致，旨在让会 SQL 而不懂 MapReduce 编程的用户可以调取 Hadoop 中的数据，进行数据处理和分析。

记录日常数据分析过程中 Hive SQL 需要的查询函数，方便手头随时查询，定期更新补充。

聚合计算

函数概览

函数语法	功能说明
`avg(x)`	平均数
`count([DISTINCT] col)`	个数，数量，记录数。count(distinct x) 为去重后数量
`sum(x)`	总数，求和
`max(x)`	最大值，日期时间字段时为最近的
`min(x)`	最小值，日期时间字段时为最早的
`collect_set(col)`	收集 group by 聚合字段，返回**去重**后数组（集合）
`collect_list(col)`	收集 group by 聚合字段，返回**不去重**的数组

函数语法	功能说明
`ntile(INTEGER x)`	用于将分组数据按照顺序切分成n片，返回当前切片值

Collect_set 样例：

分组后聚合到一个字段：

```
SELECT UID,
       collect_set(order_id)
FROM dwd.order_detail
GROUP BY UID
-- 返回类似 111 | ["123","124"]，集合的内容去重
```

用指定字符分隔内容：

```
SELECT UID,
       concat_ws(',', collect_set(cast(order_id AS string)))
FROM dwd.order_detail
GROUP BY UID
-- 返回类似 111 | 12345,45678
```

> 注：
> 1. 聚合函数需要 group by 才可使用
> 2. collect_list 不去重，collect_set 去重

字符处理

函数概览

函数语法	功能说明
`cast(expr as <type>)`	转换表达式 expr 为 type 类型
`length(int/str)`	长度
`reverse(int/str)`	反转顺序
`concat(1,2,'a')`	字符串连接
`concat_ws('-','a','b')`	指定分隔符字符串连接
`substr('foobar', 4)`	字符串截取, 或 `substring`
`substring_index(str, '-', 2)`	分隔后取前几块
`instr('abc', 'ab')`	子串的位置，0为不存在

函数语法	功能说明
`locate('a', 'abc', 1)`	子串在第 n 个位置上是否存在
`upper, ucase`	转大写
`lower,lcase`	转小写
`trim ltrim rtrim`	去空格，左右空格
`corr(col1, col2)`	
`corr(col1, col2)`	
`corr(col1, col2)`	

> todo:
> •正则表达式替换函数：regexp_replace
> •正则表达式解析函数：regexp_extract
> •URL解析函数：parse_url
> •json解析函数：get_json_object
> •空格字符串函数：space
> •重复字符串函数：repeat
> •首字符ascii函数：ascii
> •左补足函数：lpad
> •右补足函数：rpad
> •分割字符串函数: split
> •集合查找函数: find_in_set

样例：

常见的类型有：

> BIGINT，DOUBLE，FLOAT，TIMESTAMP，DATE，INTERVAL，STRING，BOOLEAN
> https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types

```
-- 返回: tom
select get_json_object('{"name":"tom"}', '$.name')
```

有效的转换结果：

```
SELECT cast(date as date) -- 返回 date 类型;
-- timestamp 中的年/月/日的值是依赖与当地的时区，结果返回 date 类型
SELECT cast(timestamp as date)
```

```
-- 如果 string 是 yyyy-MM-dd 格式的，则相应的年/月/日的 date 类型的数据将会返回；
-- 但如果 string 不是 yyyy-MM-dd 格式的，结果则会返回 NULL
SELECT cast(string as date)
-- 基于当地的时区，生成一个对应 date 的年/月/日的时间戳值；
SELECT cast(date as timestamp)
-- date所代表的年/月/日时间将会转换成 yyyy-MM-dd 的字符串。
SELECT cast(date as string)
-- 转化为 bigint
SELECT cast('1' as BIGINT)
```

字符的处理：

```
-- concat->demo
select CONCAT('concat','->','demo')
-- hive
select substr('This is hive demo',9,4)
-- 返回：abc-def
SELECT substring_index('abc-def-ghi', '-', 2)
-- 6
select length('hadoop')
-- HADOOP HADOOP
select upper('hadoop'), ucase('hadoop')
-- hadoop hadoop
select lower('HADOOP'), lcase('HADOOP')
-- HHhadoop, 保证长度，空为向左拼 H
select lpad('hadoop',8,'H')
-- hadooppp
select rpad('hadoop',8,'p')
-- HadoopHadoop
select repeat('Hadoop',2)
-- poodaH
select reverse('Hadoop')
-- ["hadoop","supports","split","function"]
select split('hadoop~supports~split~function','~')
```

正则表达式的使用：

```
-- HA^G^FER$JY 替换为 HA$G$FER$JY
select regexp_replace('HA^G^FER$JY',"\\^","\\$")
-- bar
select regexp_extract('foothebar', 'foo(.*?)(bar)', 2)
```

集合函数

函数概览

函数语法	功能说明
`array_contains(Array<T>, value)`	返回Array是否包含value
`size(Map<K.V>)`	返回Map的大小
`size(Array<T>)`	返回Array的大小
`map_keys(Map<K.V>)`	返回Map的key集合
`map_values(Map<K.V>)`	返回Map的value集合
`sort_array(Array<T>)`	返回Array是否包含value

样例：

TODO

统计运算

函数概览

函数语法	功能说明
`+, -, *, /`	加减乘除
`%`	取余数 10%3=1
`DIV`	取整数部分 17 DIV 3 = 5
`AND , OR、NOT`	和、或、非
`[NOT] IN (val1, val2, ...)`	是否在列表中有
`[NOT] EXISTS (subquery)`	只否在列表中存在
`= , <>`	等值、不等值比较
`> , <`	大于、小于比较
`>= , <=`	大于等于、小于等于比较
`IS [NOT] NULL`	空值、非空值判断
`LIKE、RLIKE`	LIKE、JAVA 的 LIKE 操作
`REGEXP`	正则表达式判断

样例：

```
-- 0 做除数（分母）时返回 NULL
select 1/0
```

> Hive正则表达式: https://www.jianshu.com/p/e81ab9a6f481

窗口函数

函数概览

函数语法	功能说明
`rank()`	相同值序号一样，跳过下个序号
`dense_rank()`	相同值序号一样，不跳过下个序号
`row_number()`	顺序排序，值同序号不同，序号不重
`cume_dist()`	同列占比，小于等于当前值的行数/分组内总行数
`lag(col,n=1,DEFAULT)`	统计窗口内往上第n行值
`lead(col,n=1,DEFAULT)`	统计窗口内往下第n行值
`first_value(col)`	分组内排序后，截止到当前行第一个值
`last_value(col)`	分组内排序后，截止到当前行最后一个值

样例：

TODO

时间函数

函数概览

函数语法	功能说明
`unix_timestamp(时间int,格式)`	指定格式日期转UNIX时间戳
`from_unixtime()`	UNIX时间戳转日期
`unix_timestamp()`	当前UNIX时间戳
`to_date()`	日期时间转日期
`date_format(time, 格式)`	对时间日期进行格式化
`year month day`	日期转年、月、日
`hour minute second`	日期转时、分、秒
`weekofyear()`	日期转周

函数语法	功能说明
`datediff()`	日期比较，时间相差
`date_add()`	日期增加
`date_sub()`	日期减少
`TRUNC（date[,fmt])`	指定元素截去日期值

样例：

```
-- 当前时间 2019-10-01 00:54:14.736
select current_timestamp()
-- 当前 日期 2019-10-01
select current_date()
-- 当前日期加一天，2019-10-02
select date_add(current_date(), 1)
-- 当前时间减一天， 2019-09-30
select date_sub(current_date(),1)
-- 当前日期所在月份的第一天，2019-10-01
select trunc(current_timestamp(), 'MONTH')
-- UNIX 时间戳转指定格式， 20190001， 'yyyymmdd' 也可以
select from_unixtime(unix_timestamp(current_date()), 'yyyyMMdd')

-- 转UNIX时间戳
SELECT unix_timestamp(20190808,'yyyyMMdd')
-- 算出日期为周几，2015-01-05 为固定值
SELECT datediff('2019-08-15', '2015-01-05')%7+1 AS week_day
-- 算出自然周数，2015-01-05 为固定值，可用于按周分组等，weekofyear() 跨年会不足一周
SELECT floor(datediff('2019-08-08', '2015-01-05')/7) AS week_number
SELECT datediff('2019-08-30', '2015-01-05') DIV 7
-- 格式化数据
SELECT date_format(time, 'yyyy-MM-dd HH:mm:ss')
-- 返回当月第一天，如 2019-09-01
select trunc(sysdate, 'mm')
-- 当前月的季度数的算法
select floor(substr('2019-09-01',6,2)/3.1)+1
```

逻辑判断

函数概览

函数语法	功能说明
`if(条件,真时值,假时值)`	条件判断
`case when`	多条件分支

函数语法	功能说明
`COALESCE(a1,a2,...,an)`	返回第一个不为 Null 的值
`isnull(a)/isnotnull(a)`	判断是否为/不为 Null
`nvl(a, b)`	a 为 Null 时返回 b, 否则为 a
`nullif(a, b)`	a = b 时，返回null，否则为a

样例：

```
-- if 语句语法，请为 Null 的设置为 0
SELECT if(var IS NULL, 0 ,var) AS var_name
-- 满足一定条件的总数，Null count() 不计数
SELECT count(if(score>=80, score, NULL)) AS good
-- case when 枚举翻译
SELECT page_id AS page_id,
      CASE page_id
          WHEN 001 THEN '第一名'
          WHEN 002 THEN '第二名'
          ELSE '-'
      END AS name
```

混合函数

Reflect 调用 java 函数

```
-- reflect (也可 java_method) 支持调用 java 自带函数，计算一行最高成绩
select reflect("java.lang.Math","max", englist, chinese) from exam
-- 返回 1 true    3   2   3   2.718281828459045   1.0
SELECT reflect("java.lang.String", "valueOf", 1),
       reflect("java.lang.String", "isEmpty"),
       reflect("java.lang.Math", "max", 2, 3),
       reflect("java.lang.Math", "min", 2, 3),
       reflect("java.lang.Math", "round", 2.5),
       reflect("java.lang.Math", "exp", 1.0),
       reflect("java.lang.Math", "floor", 1.9)
```

虚表生成

stack 堆叠

```
-- stack(INT n, v1, v2, ..., vk)，n 为列数（与 as 数对应）
SELECT stack(2, 'b', 'y', 'b2') as (b, y)
```

```sql
-- 可简少写 as
SELECT stack(1, page_id, page_name) as (page_id, page_name) FROM tt
```

explode 行转列

```sql
-- explode(列)：将列中复杂的array或者map结构数据拆分成多行
SELECT t.info_id,t.d
FROM
  (SELECT explode(info_detail) AS (info_id, detail)
   FROM my_table) as t
```

inline 列转行

```sql
-- inline(Array<Struct [, Struct]> a) 分解struct数组到表中
select *
from table1 t
  lateral view inline(array_of_structs) a;
```

其他

内置命令

```sql
show functions    -- 查看所有函数
desc function count    -- 查看函数用法
desc function extended count  -- 详细用法，简单例子
```

WITH AS

WITH AS短语，也叫做子查询部分（subquery factoring），可以定义一个SQL片断，该SQL片断会被整个SQL语句用到。

```sql
-- 相当于建了 a、b 临时表
with
     a as (select * from scott.emp),
     b as (select * from scott.dept)

select * from a, b where a.deptno = b.deptno;
```

使用总结

- 与 SQL 类似
- 代码不区分大小写
- 可以写自定义函数代码，可搜索 Hive UDF 了解

- 官方手册 https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF